Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем

и технологий»

**Курсовой проект**

по дисциплине **«Языки программирования»**

тема: Приложение **«Контроль успеваемости студентов»**

|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | студентка группы 10701122  Куликович Юлия Вячеславовна |
| Преподаватель: |  | доц. Сидорик Валерий Владимирович |

Минск 2023

Оглавление

[Введение………………………………………………………………………………….2](#_Toc103936364)

[1.Постановка и описание задачи 3](#_Toc103936365)

[2.Теоретический раздел 4](#_Toc103936366)

[3.Алгоритм. Математическая модель 5](#_Toc103936367)

[4.Практический раздел 6](#_Toc103936368)

**4.1 Структура программы......…………...….……………………………………...6**

**4.2 Описание элементов программы……………………………………………...7**

4.3 [Подключение базы данных 9](#_Toc103936371)

## **4.4 Использование PyQt6 и QtDesigner…………………………………………..10**

4.5 UML диаграммы классов……………………………………………………..11

[5. Алгоритм работы 12](#_Toc103936373)

[6. Результаты и выводы 13](#_Toc103936374)

[Приложение 1 14](#_Toc103936376)

[Список используемой литературы](#_Toc103936376) 48

# 

# **Введение**

Основные функции "Контроля успеваемости студентов" включают ввод, удаление и хранение оценок, вычисление среднего балла, а также генерацию отчетов в формате CSV для дальнейшей обработки и анализа данных.

В приложении "Контроль успеваемости студентов" уделяется особое внимание удобству использования и надежности. Оно предлагает интуитивно понятный интерфейс, который позволяет быстро и эффективно выполнять все необходимые операции. Благодаря простому в использовании механизму сохранения данных, информация оценок и успехов студентов сохраняется безопасно и доступна для будущих использования и анализа.

# **Постановка и описание задачи**

*Цель работы*: Разработать приложение «Контроль успеваемости студентов».

*Задачи:*

1. Анализ предметной области
2. Анализ литературных источников
3. Проектирование ПО
4. Реализация кода
5. Тестирование и отладка
6. Оформление курсовой работы
7. Защита курсовой работы

*Тема курсовой работы:* Разработать приложение, реализующее страницу приема заданий по дисциплине. Содержит таблицу со списком студентов и количеством заданий по дисциплине. Студенты могут только посмотреть текущее состояние дел. Преподаватель, после ввода пароля, может устанавливать факт сдачи задания. Должна быть предусмотрена возможность сортировки списка студентов по алфавиту или по рейтингу. Предусмотреть возможность проставления экзаменационной оценки студентам. Дисциплина и количество контрольных точек вводятся преподавателем. Информация хранится в базе данных.

*Детализация:* При сдаче лабораторной работы или экзамена ставится оценка от 0 до 10.

*Исходные данные*: данные, вводимые с клавиатуры.

*Специальные требования*: отсутствуют.

*Инструменты разработки*: Операционная система MsWindows ,среда разработки PyCharm, язык программирования Python.

# **Теоретический раздел**

В мире существует огромное количество университетов и дисциплин. В связи с чем возникает необходимость удобного отслеживания успеваемости студентов.

Данный проект был создан с целью облегчить и улучшить процесс отслеживания и управления успехами учащихся.

Ручное ведение учета оценок и успеваемости студентов может быть трудоемким и подверженным ошибкам. Сложно держать в голове множество оценок и их динамику для каждого ученика. Кроме того, часто требуется создание отчетов для родителей или администрации университетов, что отнимает дополнительное время и увеличивает вероятность ошибок. Приложение автоматически вычисляет средние баллы и предоставляет отчеты.

# **Алгоритм. Математическая модель**

Математическая модель описывается формулой для нахождения среднего балла по группе за экзамен:

где N – количество обучающихся, присутствовавших на аттестации

x – оценка студента

# **Практический раздел**

## **Структура программы**

Проект представлен блок-схемой или, как ее по другому называют, бизнес-логикой переключения окон проекта, состоящая из 9 окон взаимосвязанных между собой: Splash Screen (Стартовое окно приложения), About\_Author (Об авторе) , About\_Programm (О программе), About\_CSV(О csv-файлах), Entrance (Окно выбора пользователя) , Enter\_student (Окно входа для студента), Enter\_teacher (Окно входа для преподавателя), Student\_logic (Окно студента) , Teacher\_logic (Основное окно для работы с приложением, окно студента), (рис. 1.):

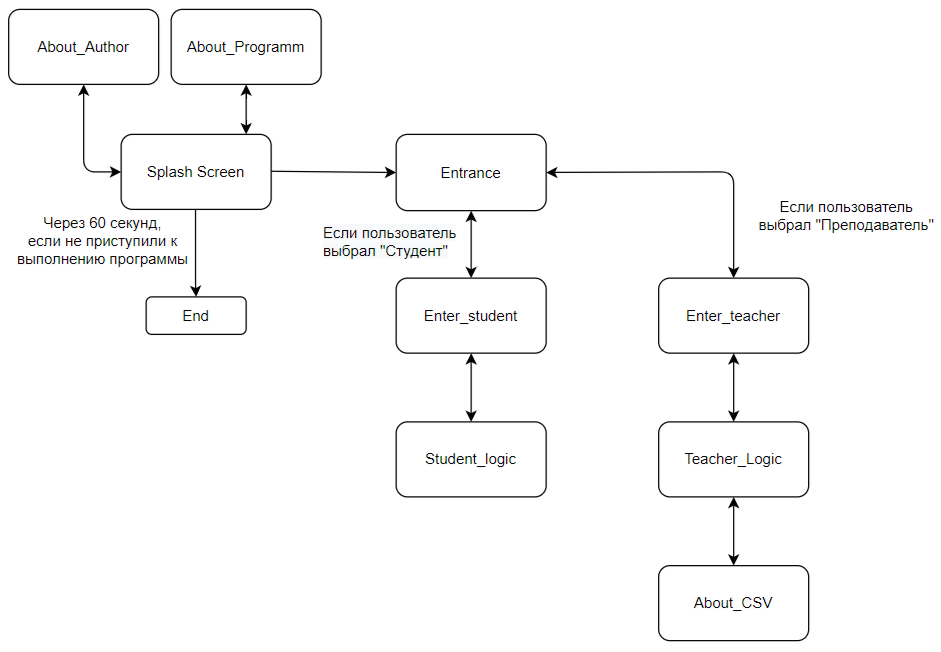


Рис 1. Бизнес-логика переключения графических окон

Структура представления материала в электронном виде рис. 2.

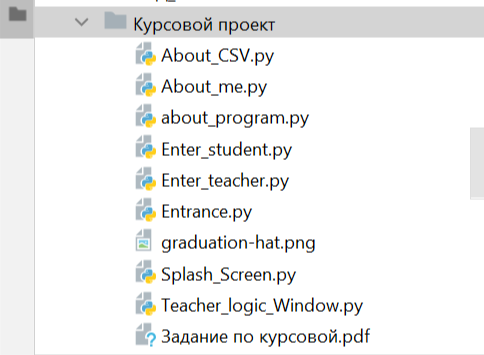


Рис. 2. Структура представления материала в электронном виде

## **Описание элементов программы**

Код данной программы см. в Приложении 1.

Программа состоит из 9 классов:

Класс SplashScreenLogic–класс, который первым запускается на выполнение. При запуске программы появляется форма, представленная в приложении 1 (рис. ). Из данной формы можно перейти на форму класса AboutProgrammLogic, представленную в приложении 1 (рис. ), AboutAuthorLogic (рис. ).

Класс Ui\_Dialog\_Entrance позволяет выбрать роль пользователю (студент или преподаватель).

Класс EnterTeacher является классом, который отвечает за вход/регистрацию для преподавателя, с заранее известным ключом доступа.

Класс StudentLogic-класс, описывающий основное окно студента, содержащее доступный ему функционал.

Класс TeacherLogic-класс, описывающий основное окно преподавателя, содержащее доступный ему функционал.

Класс AboutCSV-класс, который содержит пояснение о том, что такое csv-файлы.

Форма TeacherLogic имеет следующие элементы:

action\_save- позволяет пользователю сохранить данные с расширением .csv., содержащиеся в таблице.

action\_open- позволяет пользователю открыть данные, имеющиеся на компьютере с расширением .csv.

action\_csv- позволяет пользователю открыть окно с информацией о csv-файлах.

## **Подключение базы данных**

Для работы приложения было использовано 2 таблицы базы данных: users– хранит логин и пароль зарегистрированных пользователей, students– хранит список студентов и их отметки.

# Для создания базы данных нам нужен SQLite– свободная реляционная система управления базами данных

import sqlite3 - импортируем sqlite3 для создания и подключения к базе данных SQLite.

Создание таблицы students в окне Teacher\_Logic.py:

conn = sqlite3.connect("database\_students.db")  
cur = conn.cursor()  
cur.execute("""  
 CREATE TABLE IF NOT EXISTS students (  
 id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
 name TEXT,  
 mark1 INTEGER,  
 mark2 INTEGER,  
 mark3 INTEGER,  
 mark4 INTEGER,  
 mark5 INTEGER,  
 mark6 INTEGER);  
 """)

Создание таблицы users в окне Enter\_teacher.py:

conn = sqlite3.connect("my\_database.db")  
cursor = conn.cursor()  
cursor.execute("CREATE TABLE IF NOT EXISTS users (username TEXT, password TEXT)")

## **Использование PyQt6 и QtDesigner.**

**Qt6** - это мощный и гибкий фреймворк, разработанный для создания графических пользовательских интерфейсов (GUI) и выполнения других задач, таких как работа с базами данных, сетью, мультимедиа и многими другими. Он предоставляет набор компонентов, инструментарий для визуального проектирования интерфейсов (QtDesigner).

Пример подключения PyQt6:

from PyQt6 import QtCore, QtGui, QtWidgets

*QtCore* - модуль, предоставляющий основные классы и функциональность для работы с событиями, потоками, строками и другими базовыми компонентами Qt.

*QtGui* - модуль, содержащий классы и функциональность, связанные с графическими интерфейсами и визуальным представлением данных. Он включает в себя классы для отображения окон, виджетов, рисования, работы с изображениями, ввода и вывода текста, обработки событий мыши и клавиатуры, а также классы для работы с различными видами данных.

*QtWidgets* - модуль, предоставляющий классы и функциональность для создания пользовательского интерфейса приложений. Он содержит набор виджетов (элементов интерфейса, таких как кнопки, поля ввода, списки и др.), классы для организации графического интерфейса (например, главное окно приложения), диалоговые окна, стили оформления и многое другое.

**Qt** **Designer** - это инструмент для визуального проектирования пользовательского интерфейса (UI) во фреймворке Qt. Он предоставляет удобную среду для создания, редактирования и настройки графических элементов интерфейса вашего приложения. Можно скачать по ссылке: <https://build-system.fman.io/qt-designer-download>.(рис.3):

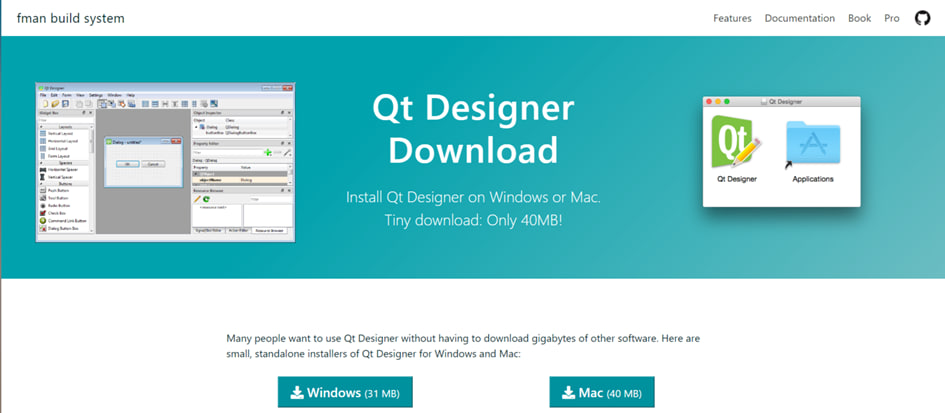


Рис. 3. Сайт, содержащий программу Qt Designer

## **UML диаграммы классов**

Ниже представлены UML диаграммы классов для приложения «Учёт посещаемости студентов» (рис. 4):

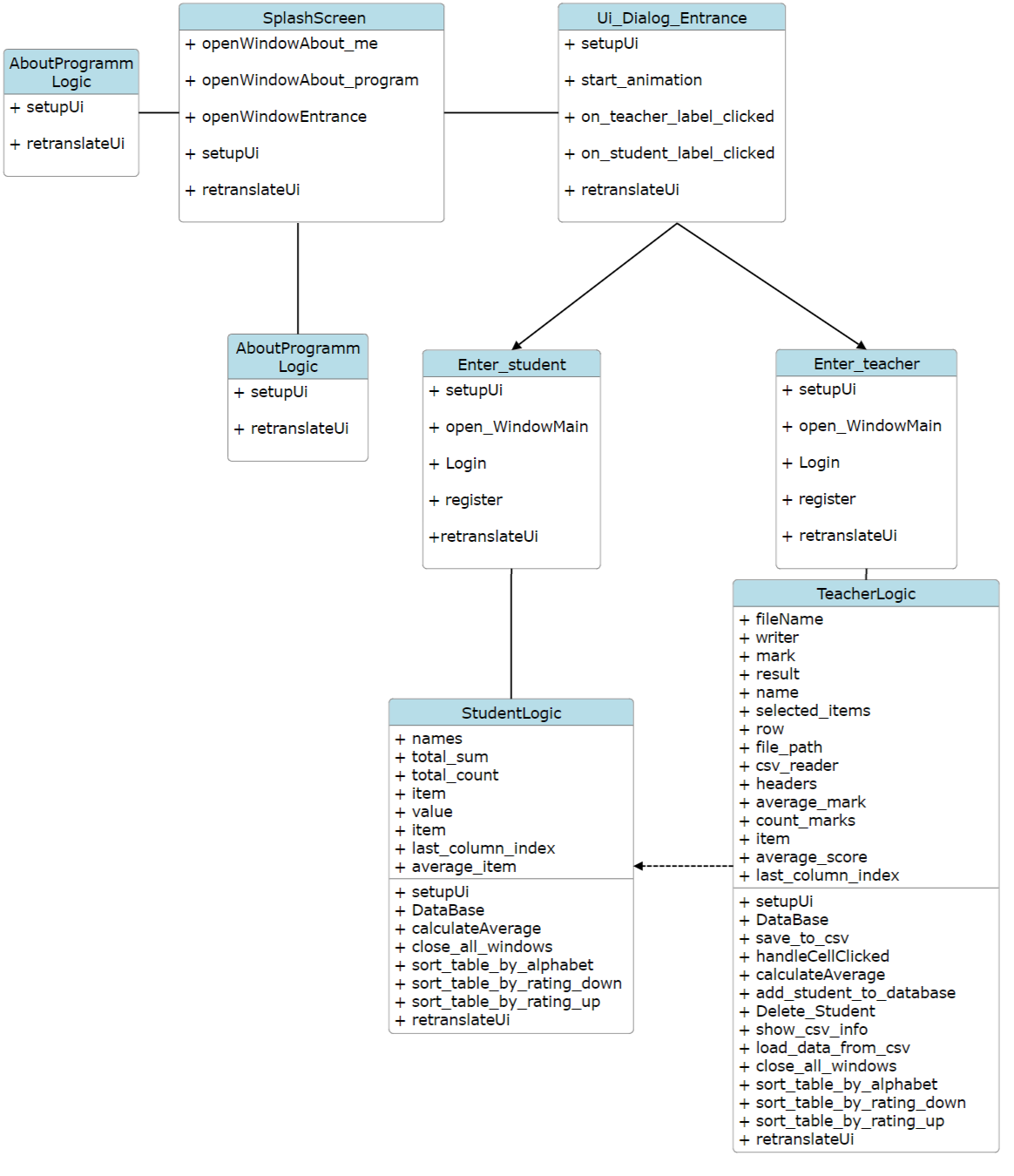


Рис. 4. UML диаграммы классов

1. **Алгоритм работы**

При запуске программы открывается стартовое окно – Splash Screen. На окне расположено 4 кнопки: при нажатии кнопки “Об авторе” откроется окно About\_author, содержащее информацию об авторе программы. При нажатии кнопки “О программе” откроется окно About Programm, содержащее информацию о программе, о том, что она выполняет. При нажатии кнопки “Выход” программа полностью завершится. При нажатии кнопки “Далее” откроется окно выбора роли пользователя (студент или преподаватель) Entrance.

После выбора роли открывается соответствующее окно входа: для преподавателя – Entrance (teacher), для студента - Entrance (student). После удачной регистрации и входа, для преподавателя открывается окно Teacher, для студента – Student.

В окне Teacher преподаватель может просмотреть успеваемость студентов в таблице, добавить студента в базу данных, удалить из базы данных, сортировать студентов по алфавиту, по возрастанию и убыванию среднего балла.

В меню можно сохранить либо открыть файл в формате csv, а также посмотреть информацию о csv-файлах, открыв окно About\_csv.

В окне Student студент может просмотреть успеваемость в таблице, сортировать студентов по алфавиту, по возрастанию и убыванию среднего балла.

# **Результаты и выводы**

Приложение предназначается для проведения процесса учёта успеваемости студентов. Данное программное решение актуально в связи с тем, что заполнение бумажной ведомости неудобно, можно допустить ошибки, а также в бумажной ведомости невозможно корректировать данные. Данное программное решение простое в использовании, здесь удобный и простой интерфейс.

# **Приложение 1**

Форма SplashScreen (рис. 5):

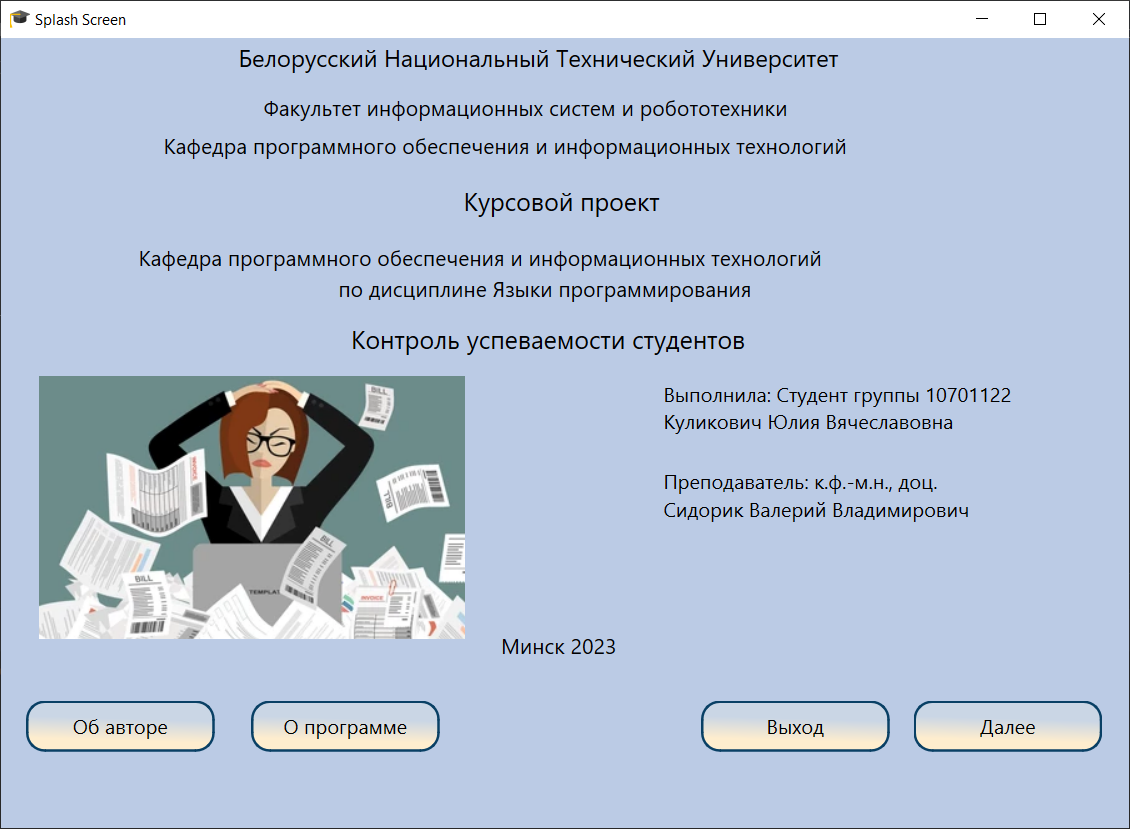


Рис. 5. Форма для класса SplashScreenLogic.py

В этом окне есть четыре кнопки: Об авторе – для перехода к окну, содержащее информацию об авторе, О программе – для перехода к окну, содержащее информацию о программе, Выход – завершение и выход из программы, Далее – кнопка для перехода к следующему окну – Entrance.

Если пользователь не нажал на кнопку Далее, программа завершит работу через 60 секунд.

Форма Entrance (рис. 6):

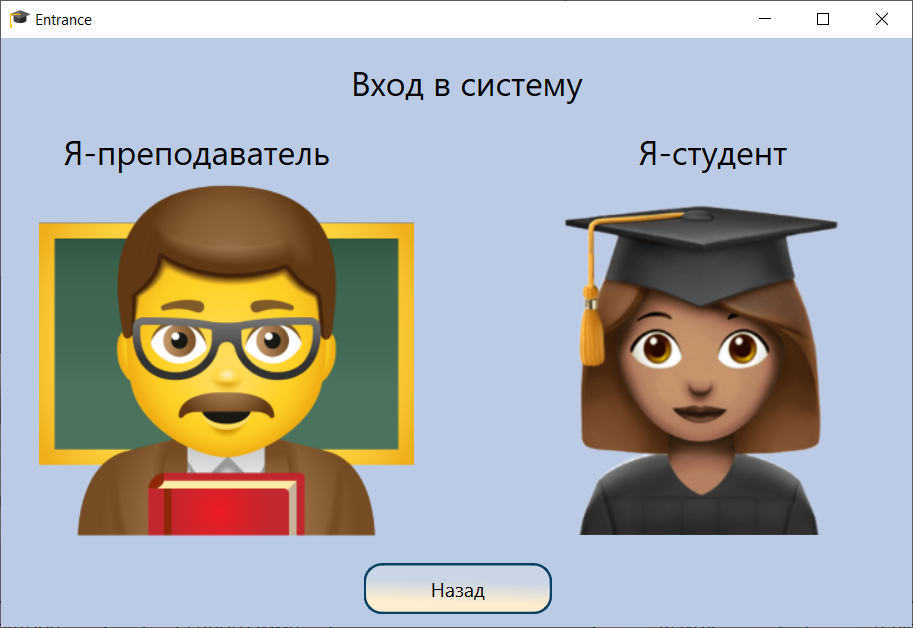


Рис. 6. Форма для класса Ui\_Dialog\_Entrance.py

Здесь содержится: выбор роли пользователя (преподаватель или студент).

Форма Enter\_teacher.py (рис. 7):

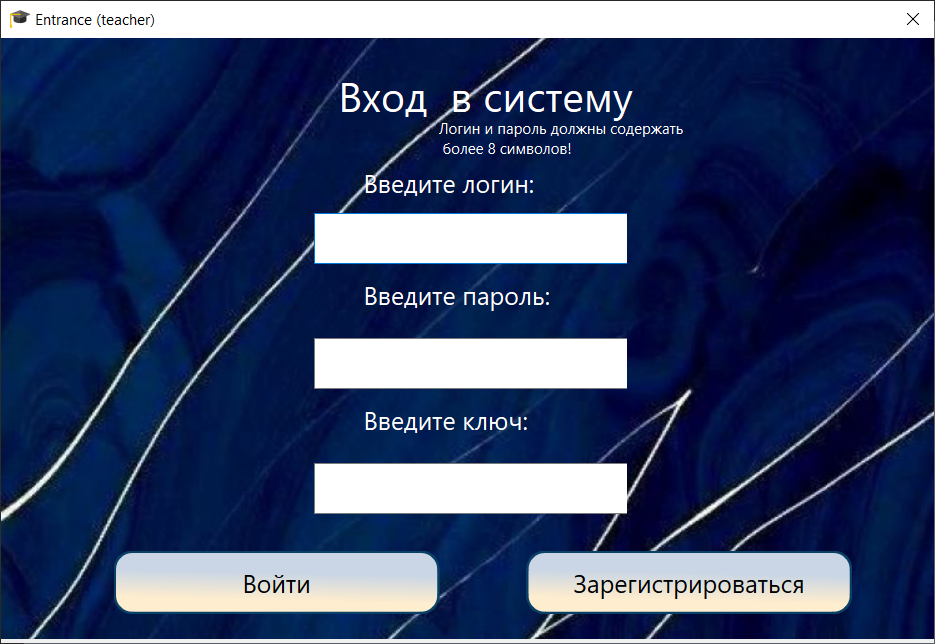


Рис. 7. Форма для класса EnterTeacher.py

Здесь содержится: ввод/регистрация преподавателем с использованием ключа доступа (доступен только преподавателю).

Форма Enter\_student.py (рис. 8):

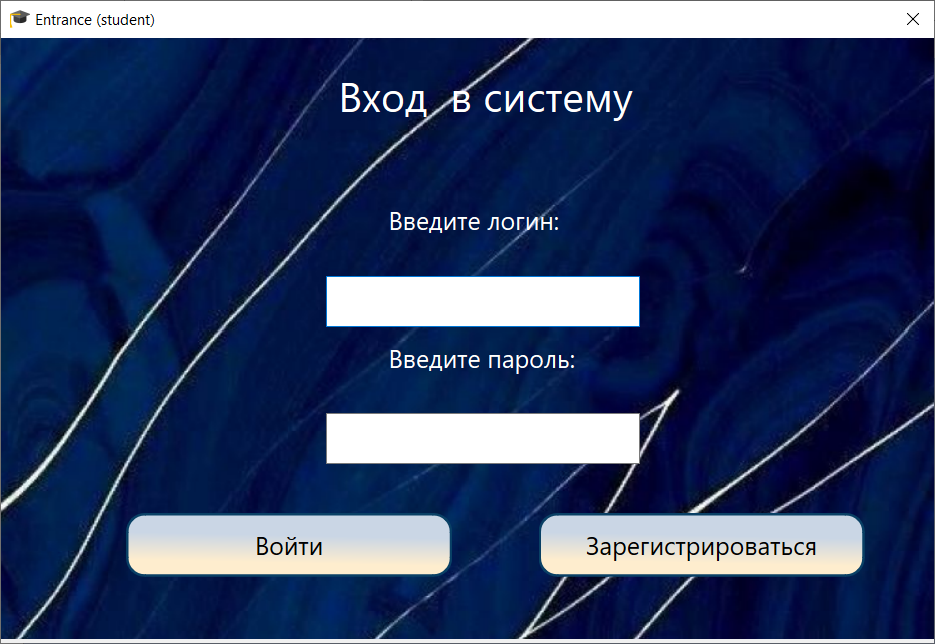


Рис. 8. Форма для класса EnterTeacher.py

Здесь содержится: ввод/регистрация студента.

Форма Teacher\_Logic.py (рис. 9):

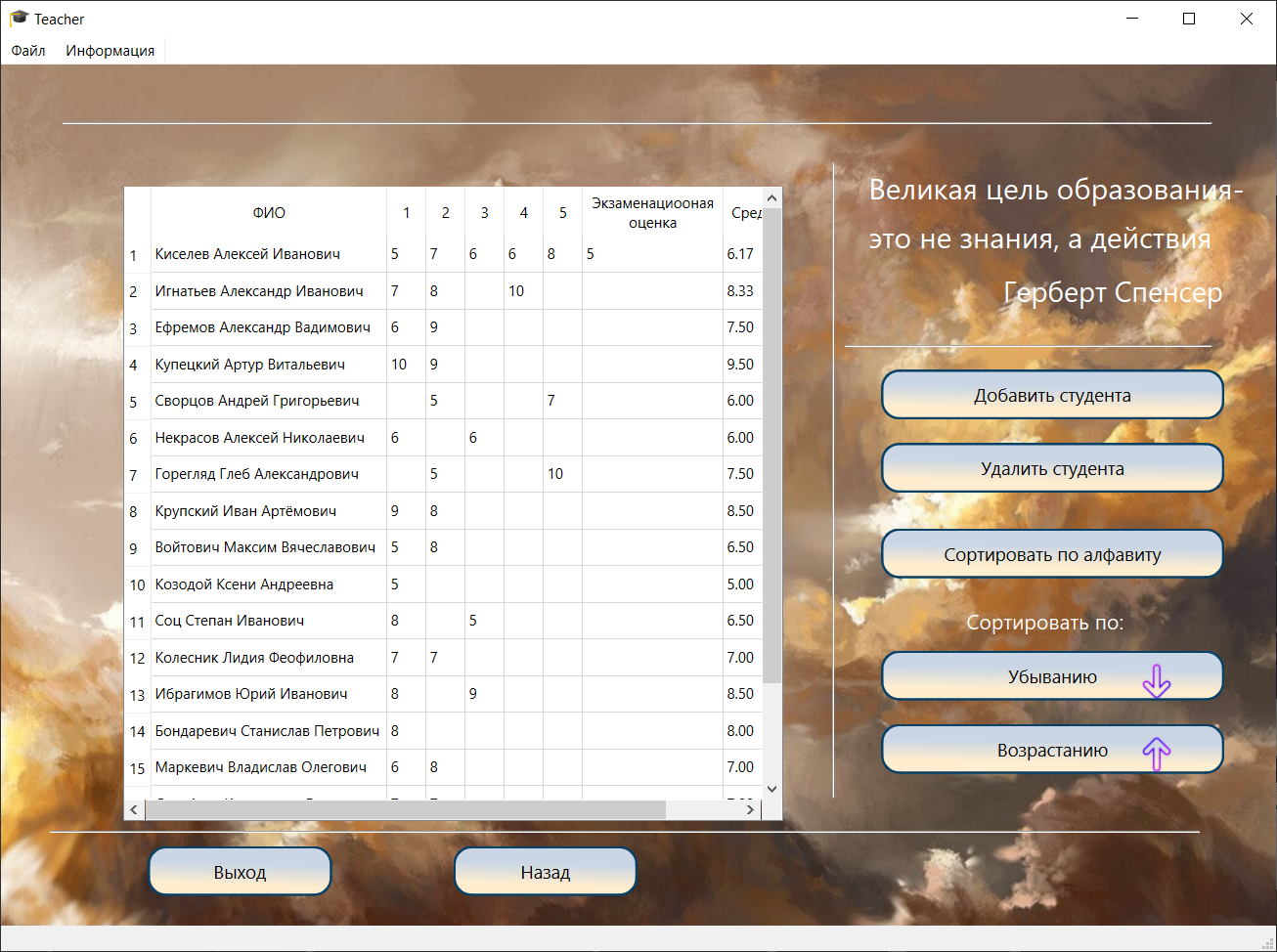


Рис. 9. Форма для класса TeacherLogic.py

Здесь содержится: таблица, содержащая список студентов, отметки за лабораторные работы, экзаменационная отметка каждого студента, их средний балл. На форме расположено 7 кнопок: Выход – для полного завершения и закрытия программы, Назад – для возврата к предыдущему окну, Добавить студента – для добавления нового студента в базу данных, Удалить студента – для удаления выбранного в таблице студента, Сортировка по алфавиту – сортировка имеющихся в таблице студентов в алфавитном порядке, Сортировка по возрастанию – сортировка студентов по возрастанию среднего балла. Сортировка по убыванию – сортировка студентов по убыванию среднего балла.

Также на форме расположено меню:

«Файл» - > «Открыть» : позволяет открыть существующий файл формата csv.

«Файл» - > «Сохранить» : позволяет сохранить существующий файл в формате csv.

«Информация» - > «О csv-файлах» : позволяет перейти к окну, содержащему информацию о csv-файлах.

Форма Student\_logic.py (рис. 8):

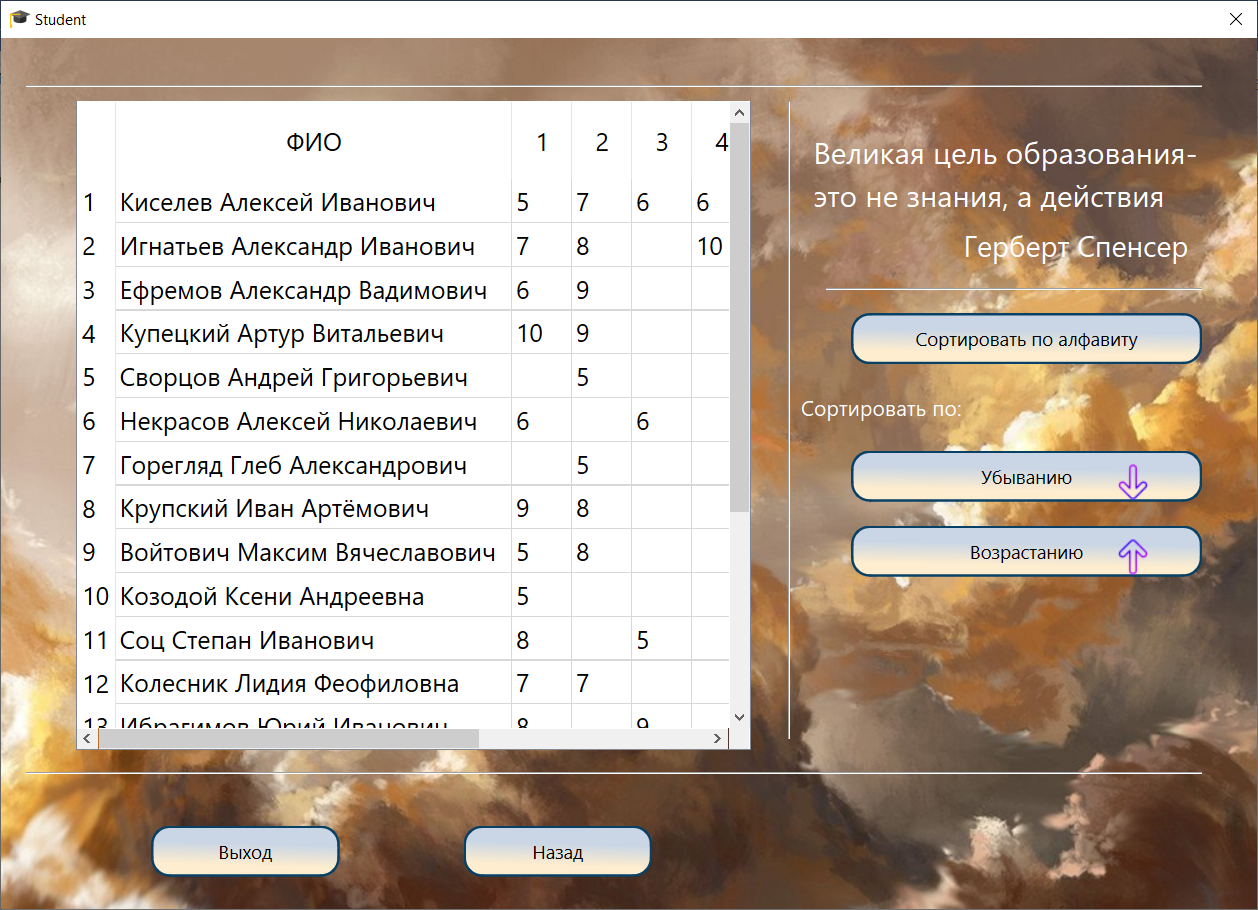


Рис. 10. Форма для класса StudentLogic.py

Здесь содержится: таблица, содержащая список студентов, отметки за лабораторные работы, экзаменационная отметка каждого студента, их средний балл. На форме расположено 5 кнопок: Выход – для полного завершения и закрытия программы, Назад – для возврата к предыдущему окну, Сортировка по алфавиту – сортировка имеющихся в таблице студентов в алфавитном порядке, Сортировка по возрастанию – сортировка студентов по возрастанию среднего балла. Сортировка по убыванию – сортировка студентов по убыванию среднего балла.

Форма About\_Author.py (рис. 11):



Рис. 11. Форма для класса AboutAuthorLogic.py

Здесь содержится: информация об авторе программы.

Форма About\_Programm.py (рис. 12):

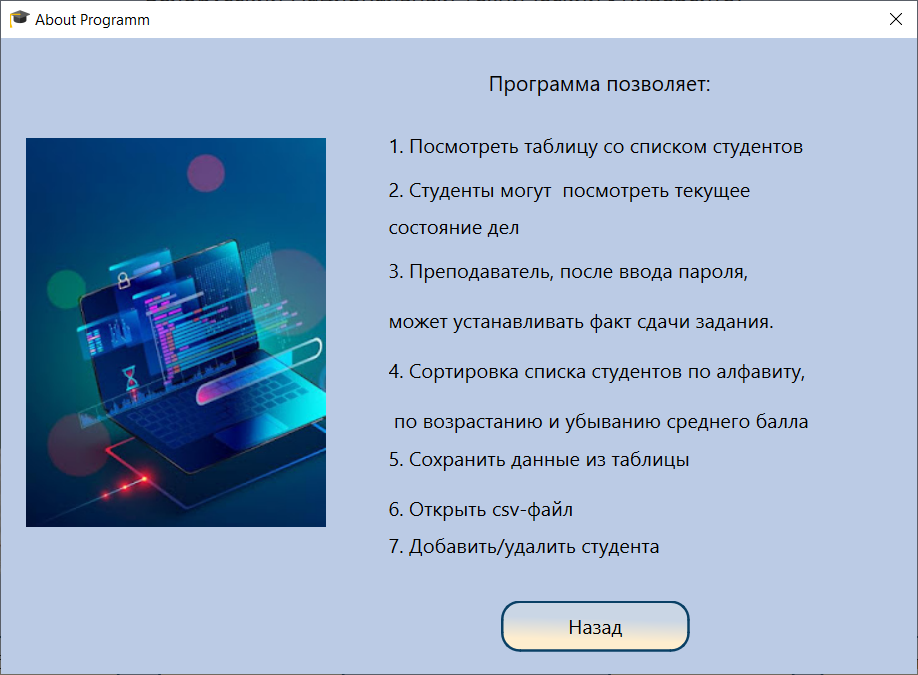


Рис. 12. Форма для класса AboutProgrammLogic.py

Здесь содержится: информация о программе.

**Код приложения:**

Стартовое окно

from PyQt6 import QtCore, QtGui, QtWidgets  
from About\_Author import AboutAuthorLogic  
from About\_Programm import AboutProgrammLogic  
from PyQt6.QtCore import QTimer  
from Entrance import Ui\_Dialog\_Entrance  
  
  
class SplashScreenLogic(object):  
 def openWindowAbout\_me(self):  
 self.window\_1 = QtWidgets.QMainWindow()  
 self.ui = AboutAuthorLogic()  
 self.ui.setupUi(self.window\_1)  
 self.window\_1.show()  
  
 def openWindowAbout\_program(self):  
 self.window\_2 = QtWidgets.QDialog()  
 self.ui = AboutProgrammLogic()  
 self.ui.setupUi(self.window\_2)  
 self.window\_2.show()  
  
 def openWindowEntrance(self):  
 self.window\_3 = QtWidgets.QMainWindow()  
 self.ui = Ui\_Dialog\_Entrance()  
 self.ui.setupUi(self.window\_3)  
 self.window\_3.show()  
  
 *# При нажатии кнопки далее останавливаем таймер* def stopTimer(self):  
 self.timer.stop()  
  
 def setupUi(self, MainWindow):  
 *# Создаем таймер* self.timer = QTimer()  
 self.timer.setSingleShot(True)  
 self.timer.timeout.connect(MainWindow.close)  
  
 self.timer.start(60000) *# Время указывается в миллисекундах (60 секунд = 60000 миллисекунд)* MainWindow.setObjectName("MainWindow")  
 MainWindow.resize(902, 632)  
 MainWindow.setStyleSheet("background-color: rgb(187, 203, 229);\n"  
"\n"  
"")  
 MainWindow.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 self.centralwidget = QtWidgets.QWidget(parent=MainWindow)  
 self.centralwidget.setObjectName("centralwidget")  
 self.label\_8 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_8.setGeometry(QtCore.QRect(530, 270, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_8.setFont(font)  
 self.label\_8.setObjectName("label\_8")  
 self.label\_2 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_2.setGeometry(QtCore.QRect(130, 70, 771, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_2.setFont(font)  
 self.label\_2.setObjectName("label\_2")  
 self.label = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label.setGeometry(QtCore.QRect(190, 0, 641, 31))  
 font = QtGui.QFont()  
 font.setPointSize(14)  
 self.label.setFont(font)  
 self.label.setObjectName("label")  
 self.label\_7 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_7.setGeometry(QtCore.QRect(280, 220, 401, 41))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_7.setFont(font)  
 self.label\_7.setObjectName("label\_7")  
 self.about\_programm = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.about\_programm.setGeometry(QtCore.QRect(200, 530, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.about\_programm.setFont(font)  
 self.about\_programm.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.about\_programm.setObjectName("pushButton\_5")  
 self.label\_10 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_10.setGeometry(QtCore.QRect(400, 470, 121, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_10.setFont(font)  
 self.label\_10.setObjectName("label\_10")  
 self.label\_9 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_9.setGeometry(QtCore.QRect(530, 340, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_9.setFont(font)  
 self.label\_9.setObjectName("label\_9")  
 self.label\_11 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_11.setGeometry(QtCore.QRect(30, 270, 341, 211))  
 self.label\_11.setText("")  
 self.label\_11.setPixmap(QtGui.QPixmap("../../Desktop/Курсовой проект/Png\_SplashScreen.webp"))  
 self.label\_11.setObjectName("label\_11")  
 self.label\_4 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_4.setGeometry(QtCore.QRect(370, 110, 231, 41))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_4.setFont(font)  
 self.label\_4.setObjectName("label\_4")  
 self.label\_5 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_5.setGeometry(QtCore.QRect(110, 160, 771, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_5.setFont(font)  
 self.label\_5.setObjectName("label\_5")  
 self.label\_3 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_3.setGeometry(QtCore.QRect(210, 40, 641, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_3.setFont(font)  
 self.label\_3.setObjectName("label\_3")  
 self.next = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.next.setGeometry(QtCore.QRect(730, 530, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.next.setFont(font)  
 self.next.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.next.setObjectName("pushButton\_4")  
 self.exxit = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.exxit.setGeometry(QtCore.QRect(560, 530, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.exxit.setFont(font)  
 self.exxit.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.exxit.setObjectName("pushButton\_2")  
 self.label\_6 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_6.setGeometry(QtCore.QRect(270, 190, 491, 21))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_6.setFont(font)  
 self.label\_6.setObjectName("label\_6")  
 self.about\_author = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.about\_author.setGeometry(QtCore.QRect(20, 530, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.about\_author.setFont(font)  
 self.about\_author.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.about\_author.setObjectName("pushButton\_3")  
 MainWindow.setCentralWidget(self.centralwidget)  
 self.menubar = QtWidgets.QMenuBar(parent=MainWindow)  
 self.menubar.setGeometry(QtCore.QRect(0, 0, 902, 26))  
 self.menubar.setObjectName("menubar")  
 MainWindow.setMenuBar(self.menubar)  
 self.statusbar = QtWidgets.QStatusBar(parent=MainWindow)  
 self.statusbar.setObjectName("statusbar")  
 MainWindow.setStatusBar(self.statusbar)  
  
 self.retranslateUi(MainWindow)  
 QtCore.QMetaObject.connectSlotsByName(MainWindow)  
  
 *# Связываем нажатие кнопок с соответствующими событиями* self.about\_author.clicked.connect(self.openWindowAbout\_me)  
 self.about\_programm.clicked.connect(self.openWindowAbout\_program)  
 self.next.clicked.connect(self.openWindowEntrance)  
 self.exxit.clicked.connect(MainWindow.close)  
 self.next.clicked.connect(self.stopTimer)  
  
 def retranslateUi(self, MainWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 MainWindow.setWindowTitle(\_translate("Dialog", "Splash Screen"))  
 self.label\_8.setText(\_translate("Dialog", "Выполнила: Студент группы 10701122\n"  
"Куликович Юлия Вячеславовна"))  
 self.label\_2.setText(\_translate("Dialog", "Кафедра программного обеспечения и информационных технологий"))  
 self.label.setText(\_translate("Dialog", "Белорусский Национальный Технический Университет"))  
 self.label\_7.setText(\_translate("Dialog", "Контроль успеваемости студентов"))  
 self.about\_programm.setText(\_translate("Dialog", "О программе"))  
 self.label\_10.setText(\_translate("Dialog", "Минск 2023"))  
 self.label\_9.setText(\_translate("Dialog", "Преподаватель: к.ф.-м.н., доц.\n"  
"Сидорик Валерий Владимирович"))  
 self.label\_4.setText(\_translate("Dialog", "Курсовой проект"))  
 self.label\_5.setText(\_translate("Dialog", "Кафедра программного обеспечения и информационных технологий"))  
 self.label\_3.setText(\_translate("Dialog", "Факультет информационных систем и робототехники"))  
 self.next.setText(\_translate("Dialog", "Далее"))  
 self.exxit.setText(\_translate("Dialog", "Выход"))  
 self.about\_author.setText(\_translate("Dialog", "Об авторе"))  
 self.label\_6.setText(\_translate("Dialog", "по дисциплине Языки программирования"))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import sys  
 app = QtWidgets.QApplication(sys.argv)  
 MainWindow = QtWidgets.QMainWindow()  
 ui = SplashScreenLogic()  
 ui.setupUi(MainWindow)  
 MainWindow.show()  
 sys.exit(app.exec())

Об авторе

from PyQt6 import QtCore, QtGui, QtWidgets  
  
  
class AboutAuthorLogic(object):  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(511, 666)  
 Dialog.setStyleSheet("background-color: rgb(187, 203, 229);")  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(110, -10, 321, 481))  
 self.label.setText("")  
 self.label.setPixmap(QtGui.QPixmap("About\_\_me.jpg"))  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(230, 470, 61, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_2.setFont(font)  
 self.label\_2.setStyleSheet("color: rgb(125, 99, 32);")  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_3.setGeometry(QtCore.QRect(130, 500, 291, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_3.setFont(font)  
 self.label\_3.setStyleSheet("color: rgb(125, 99, 32);")  
 self.label\_3.setObjectName("label\_3")  
 self.label\_4 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_4.setGeometry(QtCore.QRect(110, 540, 321, 21))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_4.setFont(font)  
 self.label\_4.setStyleSheet("color: rgb(125, 99, 32);")  
 self.label\_4.setObjectName("label\_4")  
 self.label\_5 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_5.setGeometry(QtCore.QRect(140, 570, 271, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_5.setFont(font)  
 self.label\_5.setStyleSheet("color: rgb(125, 99, 32);")  
 self.label\_5.setObjectName("label\_5")  
 self.back = QtWidgets.QPushButton(parent=Dialog)  
 self.back.setGeometry(QtCore.QRect(180, 620, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.back.setFont(font)  
 self.back.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.back.setObjectName("pushButton\_4")  
  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
 *# Закрываем окно при нажатии на кнопку* self.back.clicked.connect(Dialog.close)  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "About author"))  
 self.label\_2.setText(\_translate("Dialog", "Автор"))  
 self.label\_3.setText(\_translate("Dialog", "Студентка группы 10701122"))  
 self.label\_4.setText(\_translate("Dialog", "Куликович Юлия Вячеславовна"))  
 self.label\_5.setText(\_translate("Dialog", "yulia.kulikovich@gmail.com"))  
 self.back.setText(\_translate("Dialog", "Назад"))

О программе

from PyQt6 import QtCore, QtGui, QtWidgets  
  
  
class AboutProgrammLogic(object):  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(733, 509)  
 Dialog.setStyleSheet("background-color: rgb(187, 203, 229);\n"  
"")  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(20, 80, 241, 311))  
 self.label.setText("")  
 self.label.setPixmap(QtGui.QPixmap("about\_program.jpg"))  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(390, 20, 241, 31))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_2.setFont(font)  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_3.setGeometry(QtCore.QRect(310, 70, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_3.setFont(font)  
 self.label\_3.setObjectName("label\_3")  
 self.label\_4 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_4.setGeometry(QtCore.QRect(310, 110, 421, 21))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_4.setFont(font)  
 self.label\_4.setObjectName("label\_4")  
 self.label\_5 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_5.setGeometry(QtCore.QRect(310, 140, 191, 21))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_5.setFont(font)  
 self.label\_5.setObjectName("label\_5")  
 self.label\_6 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_6.setGeometry(QtCore.QRect(310, 170, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_6.setFont(font)  
 self.label\_6.setObjectName("label\_6")  
 self.label\_7 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_7.setGeometry(QtCore.QRect(310, 210, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_7.setFont(font)  
 self.label\_7.setObjectName("label\_7")  
 self.label\_8 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_8.setGeometry(QtCore.QRect(310, 290, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_8.setFont(font)  
 self.label\_8.setObjectName("label\_8")  
 self.label\_9 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_9.setGeometry(QtCore.QRect(310, 250, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_9.setFont(font)  
 self.label\_9.setObjectName("label\_9")  
 self.label\_10 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_10.setGeometry(QtCore.QRect(310, 320, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_10.setFont(font)  
 self.label\_10.setObjectName("label\_10")  
 self.label\_11 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_11.setGeometry(QtCore.QRect(310, 360, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_11.setFont(font)  
 self.label\_11.setObjectName("label\_11")  
 self.label\_12 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_12.setGeometry(QtCore.QRect(310, 390, 421, 31))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.label\_12.setFont(font)  
 self.label\_12.setObjectName("label\_12")  
 self.back = QtWidgets.QPushButton(parent=Dialog)  
 self.back.setGeometry(QtCore.QRect(400, 450, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.back.setFont(font)  
 self.back.setStyleSheet(" border-radius: 15px; \n"  
"border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.back.setObjectName("pushButton")  
  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
 *# Закрываем окно при нажатии на кнопку* self.back.clicked.connect(Dialog.close)  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "About Programm"))  
 self.label\_2.setText(\_translate("Dialog", "Программа позволяет:"))  
 self.label\_3.setText(\_translate("Dialog", "1. Посмотреть таблицу со списком студентов"))  
 self.label\_4.setText(\_translate("Dialog", "2. Студенты могут посмотреть текущее"))  
 self.label\_5.setText(\_translate("Dialog", "состояние дел"))  
 self.label\_6.setText(\_translate("Dialog", "3. Преподаватель, после ввода пароля,"))  
 self.label\_7.setText(\_translate("Dialog", "может устанавливать факт сдачи задания."))  
 self.label\_8.setText(\_translate("Dialog", " по возрастанию и убыванию среднего балла"))  
 self.label\_9.setText(\_translate("Dialog", "4. Cортировка списка студентов по алфавиту,"))  
 self.label\_10.setText(\_translate("Dialog", "5. Сохранить данные из таблицы"))  
 self.label\_11.setText(\_translate("Dialog", "6. Открыть csv-файл"))  
 self.label\_12.setText(\_translate("Dialog", "7. Добавить/удалить студента"))  
 self.back.setText(\_translate("Dialog", "Назад"))

Вход

from PyQt6 import QtCore, QtGui, QtWidgets  
from Enter\_teacher import EnterTeacher  
from Enter\_student import EnterStudent  
  
  
class Ui\_Dialog\_Entrance(object):  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(729, 471)  
 Dialog.setStyleSheet("background-color: rgb(187, 203, 229);\n"  
"\n"  
"")  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(280, 10, 231, 51))  
 font = QtGui.QFont()  
 font.setPointSize(20)  
 self.label.setFont(font)  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(30, 110, 331, 301))  
 self.label\_2.setStyleSheet(".scale {\n"  
" transition: 1s; /\* Время эффекта \*/\n"  
" }\n"  
" .scale:hover {\n"  
" transform: scale(1.2); /\* Увеличиваем масштаб \*/\n"  
" }")  
 self.label\_2.setText("")  
 self.label\_2.setPixmap(QtGui.QPixmap("../../Desktop/Курсовой проект/teacher.png"))  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_3.setGeometry(QtCore.QRect(450, 130, 231, 271))  
 self.label\_3.setStyleSheet(".scale {\n"  
" transition: 1s; /\* Время эффекта \*/\n"  
" }\n"  
".scale:hover {\n"  
" transform: scale(1.2); /\* Увеличиваем масштаб \*/\n"  
" }")  
 self.label\_3.setText("")  
 self.label\_3.setPixmap(QtGui.QPixmap("../../Desktop/Курсовой проект/student.png"))  
 self.label\_3.setObjectName("label\_3")  
 self.label\_4 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_4.setGeometry(QtCore.QRect(50, 70, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(20)  
 self.label\_4.setFont(font)  
 self.label\_4.setObjectName("label\_4")  
 self.label\_5 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_5.setGeometry(QtCore.QRect(510, 70, 161, 41))  
 font = QtGui.QFont()  
 font.setPointSize(20)  
 self.label\_5.setFont(font)  
 self.label\_5.setObjectName("label\_5")  
 self.back = QtWidgets.QPushButton(parent=Dialog)  
 self.back.setGeometry(QtCore.QRect(290, 420, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.back.setFont(font)  
 self.back.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.back.setObjectName("pushButton\_4")  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
  
 *# вязываем нажатие кнопок с соответствующими событиями* self.label\_2.enterEvent = self.start\_animation  
 self.label\_3.enterEvent = self.start\_animation  
 self.back.clicked.connect(Dialog.close)  
 self.label\_2.mousePressEvent = self.on\_teacher\_label\_clicked  
 self.label\_3.mousePressEvent = self.on\_student\_label\_clicked  
  
 *# При наведении на картинки отображаются всплывающие подсказки* def start\_animation(self, event):  
 self.label\_2.setToolTip("<p>Нажмите для авторизации</p>")  
 self.label\_3.setToolTip("<p>Нажмите для авторизации</p>")  
  
 def on\_teacher\_label\_clicked(self,event):  
 self.window\_7 = QtWidgets.QDialog()  
 self.ui = EnterTeacher()  
 self.ui.setupUi(self.window\_7)  
 self.window\_7.show()  
  
 def on\_student\_label\_clicked(self,event):  
 self.window\_8 = QtWidgets.QDialog()  
 self.ui = EnterStudent()  
 self.ui.setupUi(self.window\_8)  
 self.window\_8.show()  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "Entrance"))  
 self.label.setText(\_translate("Dialog", "Вход в систему"))  
 self.label\_4.setText(\_translate("Dialog", "Я-преподаватель"))  
 self.label\_5.setText(\_translate("Dialog", "Я-студент"))  
 self.back.setText(\_translate("Dialog", "Назад"))

Вход для преподавателя

from PyQt6 import QtCore, QtGui, QtWidgets  
from PyQt6.QtWidgets import QMessageBox  
from Teacher\_Logic import TeacherLogic  
import sqlite3  
  
  
class EnterTeacher(object):  
  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(746, 484)  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(0, -10, 761, 491))  
 self.label.setText("")  
 self.label.setPixmap(QtGui.QPixmap("../../Downloads/photo\_2023-11-25\_14-09-45\_\_\_2.jpg"))  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(270, 20, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(25)  
 self.label\_2.setFont(font)  
 self.label\_2.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_3.setGeometry(QtCore.QRect(290, 90, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_3.setFont(font)  
 self.label\_3.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_3.setObjectName("label\_3")  
 self.lineEdit = QtWidgets.QLineEdit(parent=Dialog)  
 self.lineEdit.setGeometry(QtCore.QRect(250, 140, 251, 41))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.lineEdit.setFont(font)  
 self.lineEdit.setObjectName("lineEdit")  
 self.label\_4 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_4.setGeometry(QtCore.QRect(290, 180, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_4.setFont(font)  
 self.label\_4.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_4.setObjectName("label\_4")  
 self.lineEdit\_2 = QtWidgets.QLineEdit(parent=Dialog)  
 self.lineEdit\_2.setGeometry(QtCore.QRect(250, 240, 251, 41))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.lineEdit\_2.setFont(font)  
 self.lineEdit\_2.setObjectName("lineEdit\_2")  
 self.lineEdit\_3 = QtWidgets.QLineEdit(parent=Dialog)  
 self.lineEdit\_3.setGeometry(QtCore.QRect(250, 340, 251, 41))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.lineEdit\_3.setFont(font)  
 self.lineEdit\_3.setObjectName("lineEdit\_3")  
 self.label\_5 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_5.setGeometry(QtCore.QRect(290, 280, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_5.setFont(font)  
 self.label\_5.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_5.setObjectName("label\_5")  
 self.enter = QtWidgets.QPushButton(parent=Dialog)  
 self.enter.setGeometry(QtCore.QRect(90, 410, 261, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.enter.setFont(font)  
 self.enter.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.enter.setObjectName("pushButton")  
 self.login = QtWidgets.QPushButton(parent=Dialog)  
 self.login.setGeometry(QtCore.QRect(420, 410, 261, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.login.setFont(font)  
 self.login.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.login.setObjectName("pushButton\_2")  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
 *# При нажатии кнопок вызываем функции Login и register* self.enter.clicked.connect(self.Login)  
 self.login.clicked.connect(self.register)  
  
 def openWindowMain(self):  
 self.window\_4 = QtWidgets.QMainWindow()  
 self.ui= TeacherLogic()  
 self.ui.setupUi(self.window\_4)  
 self.window\_4.show()  
  
 def Login(self):  
 username = self.lineEdit.text()  
 password = self.lineEdit\_2.text()  
  
 msg = QMessageBox()  
 msg.setIcon(QMessageBox.Icon.Warning)  
  
 key = self.lineEdit\_3.text().strip()  
  
 if not username or not password: *# Check if either username or password field is empty* msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, не заполнены все поля!")  
 msg.exec()  
 elif len(username) <= 8 or len(password) <= 8: *# Check if username or password has less than 8 characters* msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, имя пользователя или пароль должны содержать более 8 символов!")  
 msg.exec()  
 else:  
 if key == "123":  
 conn = sqlite3.connect("my\_database.db")  
 cursor = conn.cursor()  
 *# Проверяем, верно ли введены данные* cursor.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (username, password))  
 result = cursor.fetchone()  
  
 if result:  
 *# Открываем окно преподавателя* self.openWindowMain()  
 else:  
 msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, неверно введён логин или пароль!")  
 msg.exec()  
 else:  
 msg.setText("Введён неверный ключ!")  
 msg.setWindowTitle("Ошибка")  
 msg.exec()  
  
 def register(self):  
 conn = sqlite3.connect("my\_database.db")  
 cursor = conn.cursor()  
 cursor.execute("CREATE TABLE IF NOT EXISTS users (username TEXT, password TEXT)")  
  
 username = self.lineEdit.text()  
 password = self.lineEdit\_2.text()  
  
 msg = QMessageBox()  
 msg.setIcon(QMessageBox.Icon.Warning)  
  
 if not username or not password:  
 msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, не заполнены все поля!")  
 msg.exec()  
 elif len(username) <= 8 or len(password) <= 8:  
 msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, имя пользователя или пароль должны содержать более 8 символов!")  
 msg.exec()  
 else:  
 *# Сверяем ключ доступа* key = self.lineEdit\_3.text().strip()  
 if key == "123":  
 cursor.execute("INSERT INTO users VALUES (?, ?)", (username, password))  
 conn.commit()  
 msg.setWindowTitle("Успех")  
 msg.setText("Вы успешно были зарегистрированы!")  
 msg.exec()  
 else:  
 msg.setWindowTitle("Ошибка")  
 msg.setText("Введён неверный ключ!")  
 msg.exec()  
  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "Entrance (teacher)"))  
 self.label\_2.setText(\_translate("Dialog", "Вход в систему"))  
 self.label\_3.setText(\_translate("Dialog", "Введите логин:"))  
 self.label\_4.setText(\_translate("Dialog", "Введите пароль:"))  
 self.label\_5.setText(\_translate("Dialog", "Введите ключ:"))  
 self.enter.setText(\_translate("Dialog", "Войти"))  
 self.login.setText(\_translate("Dialog", "Зарегистрироваться"))  
  
 self.label\_6 = QtWidgets.QLabel(Dialog)  
 self.label\_6.setGeometry(QtCore.QRect(350, 60, 400, 40))  
 self.label\_6.setObjectName("label\_6")  
 self.label\_6.setText(\_translate("Dialog", "Логин и пароль должны содержать\n более 8 символов!"))  
 self.label\_6.setStyleSheet("color: rgb(255, 255, 255);")

Вход для студента

from PyQt6 import QtCore, QtGui, QtWidgets  
*# Импортируем класс StudentLogic из Student\_logic.ру*from Student\_logic import StudentLogic  
from PyQt6.QtWidgets import QMessageBox  
import sqlite3  
  
  
class EnterStudent(object):  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(746, 484)  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(-10, -10, 761, 491))  
 self.label.setText("")  
 self.label.setPixmap(QtGui.QPixmap("../../Downloads/photo\_2023-11-25\_14-09-45\_\_\_2.jpg"))  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(270, 20, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(25)  
 self.label\_2.setFont(font)  
 self.label\_2.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_3.setGeometry(QtCore.QRect(310, 120, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_3.setFont(font)  
 self.label\_3.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_3.setObjectName("label\_3")  
 self.lineEdit = QtWidgets.QLineEdit(parent=Dialog)  
 self.lineEdit.setGeometry(QtCore.QRect(260, 190, 251, 41))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.lineEdit.setFont(font)  
 self.lineEdit.setObjectName("lineEdit")  
 self.label\_4 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_4.setGeometry(QtCore.QRect(310, 230, 361, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_4.setFont(font)  
 self.label\_4.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_4.setObjectName("label\_4")  
 self.lineEdit\_2 = QtWidgets.QLineEdit(parent=Dialog)  
 self.lineEdit\_2.setGeometry(QtCore.QRect(260, 300, 251, 41))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.lineEdit\_2.setFont(font)  
 self.lineEdit\_2.setObjectName("lineEdit\_2")  
 self.enter = QtWidgets.QPushButton(parent=Dialog)  
 self.enter.setGeometry(QtCore.QRect(100, 380, 261, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.enter.setFont(font)  
 self.enter.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.enter.setObjectName("pushButton")  
 self.registr = QtWidgets.QPushButton(parent=Dialog)  
 self.registr.setGeometry(QtCore.QRect(430, 380, 261, 51))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.registr.setFont(font)  
 self.registr.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.registr.setObjectName("pushButton\_2")  
  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
  
 *# При нажатии кнопок enter и registr, вызываем ф-ии Login и Register* self.enter.clicked.connect(self.Login)  
 self.registr.clicked.connect(self.Register)  
  
 def Login(self):  
 *# Получаем логин и пароль из полей для ввода* username = self.lineEdit.text()  
 password = self.lineEdit\_2.text()  
  
 msg = QMessageBox()  
 msg.setIcon(QMessageBox.Icon.Warning)  
 if not username or not password: *#* msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, не заполнены все поля!")  
 msg.exec()  
 else:  
 conn = sqlite3.connect("my\_database.db")  
 cursor = conn.cursor()  
 *# Сверяем введенные данные с теми, что хранятся в бд* cursor.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (username, password))  
 result = cursor.fetchone()  
  
 if result:  
 *# открываем окно для студента* self.openWindowMain()  
 else:  
 msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, неверно введён логин или пароль!")  
 msg.exec()  
  
 def openWindowMain(self):  
 self.window\_9 = QtWidgets.QDialog()  
 self.ui= StudentLogic()  
 self.ui.setupUi(self.window\_9)  
 self.window\_9.show()  
  
 def Register(self):  
 conn = sqlite3.connect("my\_database.db")  
 cursor = conn.cursor()  
 cursor.execute("CREATE TABLE IF NOT EXISTS users (username TEXT, password TEXT)")  
  
 username = self.lineEdit.text()  
 password = self.lineEdit\_2.text()  
  
 msg = QMessageBox()  
 msg.setIcon(QMessageBox.Icon.Warning)  
 if not username or not password:  
 msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, не заполнены все поля!")  
 msg.exec()  
 elif len(username) <= 8 or len(password) <= 8: *# Check if username or password has less than 8 characters* msg.setWindowTitle("Ошибка")  
 msg.setText("Ошибка, имя пользователя или пароль должны содержать более 8 символов!")  
 msg.exec()  
 else:  
 cursor.execute("INSERT INTO users VALUES (?, ?)", (username, password))  
 conn.commit()  
 msg.setWindowTitle("Успех")  
 msg.setText("Вы успешно были зарегистрированы!")  
 msg.exec()  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "Entrance (student)"))  
 self.label\_2.setText(\_translate("Dialog", "Вход в систему"))  
 self.label\_3.setText(\_translate("Dialog", "Введите логин:"))  
 self.label\_4.setText(\_translate("Dialog", "Введите пароль:"))  
 self.enter.setText(\_translate("Dialog", "Войти"))  
 self.registr.setText(\_translate("Dialog", "Зарегистрироваться"))

Окно преподавателя

from PyQt6 import QtCore, QtGui, QtWidgets  
from PyQt6.QtWidgets import QFileDialog, QMessageBox  
import sqlite3  
import csv  
from About\_CSV import AboutCSV  
from PyQt6.QtCore import Qt  
  
  
class TeacherLogic(object):  
  
 def setupUi(self, MainWindow):  
 MainWindow.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 MainWindow.setObjectName("MainWindow")  
 MainWindow.resize(1043, 748)  
 self.centralwidget = QtWidgets.QWidget(parent=MainWindow)  
 self.centralwidget.setObjectName("centralwidget")  
 self.label = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label.setGeometry(QtCore.QRect(0, -10, 1041, 661))  
 self.label.setText("")  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_2.setGeometry(QtCore.QRect(-20, -60, 1071, 871))  
 self.label\_2.setText("")  
 self.label\_2.setPixmap(QtGui.QPixmap("../../Downloads/1678348032\_bogatyr-club-p-tuchi-estetika-foni-pinterest-57.jpg"))  
 self.label\_2.setObjectName("label\_2")  
 self.line\_2 = QtWidgets.QFrame(parent=self.centralwidget)  
 self.line\_2.setGeometry(QtCore.QRect(50, 40, 941, 16))  
 self.line\_2.setFrameShape(QtWidgets.QFrame.Shape.HLine)  
 self.line\_2.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line\_2.setObjectName("line\_2")  
 self.line = QtWidgets.QFrame(parent=self.centralwidget)  
 self.line.setGeometry(QtCore.QRect(670, 80, 20, 521))  
 self.line.setFrameShape(QtWidgets.QFrame.Shape.VLine)  
 self.line.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line.setObjectName("line")  
 self.line\_4 = QtWidgets.QFrame(parent=self.centralwidget)  
 self.line\_4.setGeometry(QtCore.QRect(690, 220, 301, 20))  
 self.line\_4.setFrameShape(QtWidgets.QFrame.Shape.HLine)  
 self.line\_4.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line\_4.setObjectName("line\_4")  
 self.label\_3 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_3.setGeometry(QtCore.QRect(710, 70, 311, 61))  
 font = QtGui.QFont()  
 font.setPointSize(18)  
 self.label\_3.setFont(font)  
 self.label\_3.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_3.setObjectName("label\_3")  
 self.label\_5 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_5.setGeometry(QtCore.QRect(820, 170, 221, 31))  
 font = QtGui.QFont()  
 font.setPointSize(17)  
 self.label\_5.setFont(font)  
 self.label\_5.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_5.setObjectName("label\_5")  
 self.add\_student = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.add\_student.setGeometry(QtCore.QRect(720, 250, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.add\_student.setFont(font)  
 self.add\_student.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.add\_student.setObjectName("pushButton\_7")  
 self.delete\_student = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.delete\_student.setGeometry(QtCore.QRect(720, 310, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.delete\_student.setFont(font)  
 self.delete\_student.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.delete\_student.setObjectName("pushButton\_8")  
 self.sorted\_alphabet = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.sorted\_alphabet.setGeometry(QtCore.QRect(720, 380, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.sorted\_alphabet.setFont(font)  
 self.sorted\_alphabet.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.sorted\_alphabet.setObjectName("pushButton\_5")  
 self.sorted\_down = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.sorted\_down.setGeometry(QtCore.QRect(720, 480, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.sorted\_down.setFont(font)  
 self.sorted\_down.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.sorted\_down.setObjectName("pushButton\_6")  
 self.label\_4 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_4.setGeometry(QtCore.QRect(790, 430, 171, 51))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_4.setFont(font)  
 self.label\_4.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_4.setObjectName("label\_4")  
 self.sorted\_up = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.sorted\_up.setGeometry(QtCore.QRect(720, 540, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.sorted\_up.setFont(font)  
 self.sorted\_up.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.sorted\_up.setObjectName("pushButton\_9")  
 self.label\_6 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_6.setGeometry(QtCore.QRect(930, 490, 61, 31))  
 self.label\_6.setText("")  
 self.label\_6.setPixmap(QtGui.QPixmap("../../Downloads/Стрелка вниз.png"))  
 self.label\_6.setObjectName("label\_6")  
 self.label\_7 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_7.setGeometry(QtCore.QRect(930, 550, 61, 31))  
 self.label\_7.setText("")  
 self.label\_7.setPixmap(QtGui.QPixmap("../../Downloads/Стрелка\_вверх.png"))  
 self.label\_7.setObjectName("label\_7")  
 self.exit = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.exit.setGeometry(QtCore.QRect(120, 640, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.exit.setFont(font)  
 self.exit.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.exit.setObjectName("pushButton\_10")  
 self.back = QtWidgets.QPushButton(parent=self.centralwidget)  
 self.back.setGeometry(QtCore.QRect(370, 640, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.back.setFont(font)  
 self.back.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.back.setObjectName("pushButton\_12")  
 self.line\_3 = QtWidgets.QFrame(parent=self.centralwidget)  
 self.line\_3.setGeometry(QtCore.QRect(40, 620, 941, 16))  
 self.line\_3.setFrameShape(QtWidgets.QFrame.Shape.HLine)  
 self.line\_3.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line\_3.setObjectName("line\_3")  
 self.label\_8 = QtWidgets.QLabel(parent=self.centralwidget)  
 self.label\_8.setGeometry(QtCore.QRect(710, 110, 311, 61))  
 font = QtGui.QFont()  
 font.setPointSize(18)  
 self.label\_8.setFont(font)  
 self.label\_8.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_8.setObjectName("label\_8")  
 MainWindow.setCentralWidget(self.centralwidget)  
 self.menubar = QtWidgets.QMenuBar(parent=MainWindow)  
 self.menubar.setGeometry(QtCore.QRect(0, 0, 1043, 26))  
 self.menubar.setObjectName("menubar")  
 self.menu\_file = QtWidgets.QMenu(parent=self.menubar)  
 self.menu\_file.setObjectName("menu")  
 self.menu\_information = QtWidgets.QMenu(parent=self.menubar)  
 self.menu\_information.setObjectName("menu\_2")  
 MainWindow.setMenuBar(self.menubar)  
 self.statusbar = QtWidgets.QStatusBar(parent=MainWindow)  
 self.statusbar.setObjectName("statusbar")  
 MainWindow.setStatusBar(self.statusbar)  
 self.action\_save = QtGui.QAction(parent=MainWindow)  
 self.action\_save.setObjectName("action\_3")  
 self.action\_open = QtGui.QAction(parent=MainWindow)  
 self.action\_open.setObjectName("action\_4")  
 self.action\_csv = QtGui.QAction(parent=MainWindow)  
 self.action\_csv.setObjectName("action\_csv")  
 self.menu\_file.addSeparator()  
 self.menu\_file.addSeparator()  
 self.menu\_file.addAction(self.action\_save)  
 self.menu\_file.addAction(self.action\_open)  
 self.menu\_information.addAction(self.action\_csv)  
 self.menubar.addAction(self.menu\_file.menuAction())  
 self.menubar.addAction(self.menu\_information.menuAction())  
  
 self.retranslateUi(MainWindow)  
 QtCore.QMetaObject.connectSlotsByName(MainWindow)  
  
 self.DataBase()  
 self.exit.clicked.connect(self.close\_all\_windows)  
 self.sorted\_alphabet.clicked.connect(self.sort\_table\_by\_alphabet)  
 self.sorted\_up.clicked.connect(self.sort\_table\_by\_rating\_up)  
 self.add\_student.clicked.connect(self.add\_student\_to\_database)  
 self.delete\_student.clicked.connect(self.Delete\_Student)  
 self.back.clicked.connect(MainWindow.close)  
 self.sorted\_down.clicked.connect(self.sort\_table\_by\_rating\_down)  
 self.students.cellClicked.connect(self.handleCellClicked)  
 self.calculateAverage()  
 self.action\_csv.triggered.connect(self.show\_csv\_info)  
 self.action\_open.triggered.connect(self.load\_data\_from\_csv)  
 self.action\_save.triggered.connect(self.save\_to\_csv)  
  
 *# Создаем пустой список для хранения фио* def DataBase(self):  
 self.students = QtWidgets.QTableWidget(self.centralwidget)  
 self.students.setGeometry(QtCore.QRect(100, 100, 540, 520))  
 self.students.setObjectName("students")  
 self.students.setColumnCount(8)  
 self.students.setRowCount(0)  
 self.students.raise\_()  
 self.students.setHorizontalHeaderLabels(['ФИО'] + [str(i) for i in range(1, 6)] + ["Экзаменациооная\nоценка"]+["Средний балл"])  
  
 names = []  
  
 *# Создаём таблицу, если еще не существует* conn = sqlite3.connect("database\_students.db")  
 cur = conn.cursor()  
 cur.execute("""  
 CREATE TABLE IF NOT EXISTS students (  
 id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
 name TEXT,  
 mark1 INTEGER,  
 mark2 INTEGER,  
 mark3 INTEGER,  
 mark4 INTEGER,  
 mark5 INTEGER,  
 mark6 INTEGER);  
 """)  
  
 for name in names:  
 *# Здесь мы сравниваем значение столбца name таблицы "students" с переданным значением name переменной* cur.execute("""INSERT OR IGNORE INTO students (name)  
 SELECT ?  
 WHERE NOT EXISTS (  
 SELECT name FROM students WHERE name = ?);""", (name, name,))  
 *# Выбираем все значения из столбца name и сохранем в переменную names* cur.execute("SELECT name FROM students")  
  
 cur.execute("SELECT name, mark1, mark2, mark3, mark4, mark5, mark6 FROM students")  
 rows = cur.fetchall()  
  
 self.students.setRowCount(len(rows))  
 for row, data in enumerate(rows):  
 for col, value in enumerate(data):  
 if value is not None:  
 item = QtWidgets.QTableWidgetItem(str(value))  
 else:  
 item = QtWidgets.QTableWidgetItem("") *# Создаем пустой элемент ячейки* self.students.setItem(row, col, item)  
 self.students.resizeColumnsToContents()  
 conn.commit()  
 conn.close()  
  
 def save\_to\_csv(self):  
 *# Открывается диалог сохранения файла CSV* fileName, \_ = QFileDialog.getSaveFileName(None, "Сохранить как CSV", "", "CSV Files (\*.csv)")  
 if fileName:  
 conn = sqlite3.connect("database\_students.db")  
 cur = conn.cursor()  
 *# Выбираются необходимые данные из базы данных* cur.execute("SELECT name, mark1, mark2, mark3, mark4, mark5, mark6 FROM students")  
 data = cur.fetchall()  
  
 *# Открывается файл для записи данных* with open(fileName, 'w', newline='', encoding='utf-8-sig') as file:  
 *# Создается объект для записи CSV* writer = csv.writer(file, delimiter=';')  
 writer.writerow(  
 ['ФИО', 'Оценка 1', 'Оценка 2', 'Оценка 3', 'Оценка 4', 'Оценка 5', 'Экзаменационная отметка'])  
  
 *# Записываются данные из базы данных* writer.writerows(data)  
 conn.close()  
  
 def calculateAverage(self):  
 for row in range(self.students.rowCount()):  
 total\_sum = 0  
 total\_count = 0  
 for col in range(1, 9):  
 item = self.students.item(row, col)  
 if item is not None and item.text():  
 value = float(item.text())  
 total\_sum += value  
 total\_count += 1  
 if total\_count != 0:  
 average = total\_sum / total\_count  
 last\_column\_index = self.students.columnCount() - 1  
 average\_item = QtWidgets.QTableWidgetItem(f"{average:.2f}")  
 self.students.setItem(row, last\_column\_index, average\_item)  
  
 def add\_student\_to\_database(self):  
 name, ok = QtWidgets.QInputDialog.getText(None, "Добавить студента", "Введите ФИО студента:")  
 if ok:  
 if name:  
 msg = QMessageBox()  
 msg.setIcon(QMessageBox.Icon.Warning)  
 if len(name.split()) == 3:  
 row\_num = self.students.rowCount()  
 self.students.insertRow(row\_num)  
 name\_item = QtWidgets.QTableWidgetItem(name) *# используется для хранения информации виджета таблицы* self.students.setItem(row\_num, 0, name\_item)  
 try:  
 connection = sqlite3.connect('database\_students.db')  
 cursor = connection.cursor()  
 cursor.execute("INSERT INTO students (name) VALUES (?)", (name,))  
 connection.commit()  
 connection.close()  
 except Exception as e:  
 msg.setText(f"Ошибка при попытке подключения к базе данных: {e}")  
 msg.setWindowTitle("Ошибка")  
 msg.exec()  
 else:  
 msg.setText("Будьте внимательны! Введите ФИО ученика в формате 'Фамилия Имя Отчество'.")  
 msg.setWindowTitle("Ошибка")  
 msg.exec()  
 else:  
 msg = QMessageBox()  
 msg.setIcon(QMessageBox.Icon.Warning)  
 msg.setText("Поле не заполнено. Введите ФИО студента.")  
 msg.setWindowTitle("Ошибка")  
 msg.exec()  
  
 def Delete\_Student(self):  
 selected\_items = self.students.selectedItems()  
 if selected\_items:  
 row = selected\_items[0].row()  
 name = self.students.item(row, 0).text()  
 *# Отображение диалогового окна подтверждения* message\_box = QMessageBox(self.students)  
 message\_box.setWindowTitle("Удаление студента")  
 message\_box.setText(f"Вы уверены, что хотите удалить студента {name}?")  
 message\_box.setIcon(QMessageBox.Icon.Question)  
 message\_box.setStandardButtons(QMessageBox.StandardButton.Yes | QMessageBox.StandardButton.No)  
 message\_box.setDefaultButton(QMessageBox.StandardButton.No)  
 *# Проверка результата диалогового окна* result = message\_box.exec()  
  
 if result == QMessageBox.StandardButton.Yes:  
 conn = sqlite3.connect("database\_students.db")  
 cur = conn.cursor()  
 cur.execute("DELETE FROM students WHERE name=?", (name,))  
 conn.commit()  
 conn.close()  
 self.students.removeRow(row)  
  
 def close\_all\_windows(self):  
 app = QtWidgets.QApplication.instance()  
 for widget in app.topLevelWidgets():  
 if widget.isVisible():  
 widget.close()  
  
 def show\_csv\_info(self):  
 self.window\_3 = QtWidgets.QMainWindow()  
 self.ui = AboutCSV()  
 self.ui.setupUi(self.window\_3)  
 self.window\_3.show()  
  
 def load\_data\_from\_csv(self):  
 file\_path, \_ = QtWidgets.QFileDialog.getOpenFileName(None, "Выберите CSV-файл", "", "CSV Files (\*.csv)")  
 if file\_path:  
 with open(file\_path, 'r', encoding='utf-8') as csvfile:  
 csv\_reader = csv.reader(csvfile, delimiter=';')  
 headers = next(csv\_reader)  
 self.students.setColumnCount(len(headers) + 1) *# Установить количество столбцов на единицу больше* self.students.setHorizontalHeaderLabels(  
 headers + ["Средний балл"]) *# Добавить заголовок для среднего балла* self.students.setRowCount(0)  
 for row\_data in csv\_reader:  
 row\_num = self.students.rowCount()  
 self.students.insertRow(row\_num)  
 average\_mark = 0  
 count\_marks = 0  
 for col\_num, col\_data in enumerate(row\_data):  
 item = QtWidgets.QTableWidgetItem(col\_data)  
 if col\_num > 0:  
 if col\_data:  
 average\_mark += int(col\_data)  
 count\_marks += 1  
 self.students.setItem(row\_num, col\_num, item)  
  
 if count\_marks > 0:  
 average\_score = round(average\_mark / count\_marks, 2)  
 else:  
 average\_score = 0  
  
 item = QtWidgets.QTableWidgetItem(str(average\_score))  
 item.setData(QtCore.Qt.ItemDataRole.DisplayRole, f'{average\_score:.2f}')  
 self.students.setItem(row\_num, len(row\_data), item)  
 self.students.resizeColumnsToContents()  
 self.calculateAverage()  
  
 def sort\_table\_by\_alphabet(self, column):  
 self.students.sortItems(column)  
  
 def sort\_table\_by\_rating\_down(self, column):  
 last\_column\_index = self.students.columnCount() - 1  
 self.students.sortItems(last\_column\_index, Qt.SortOrder.DescendingOrder)  
  
 def sort\_table\_by\_rating\_up(self, column):  
 last\_column\_index = self.students.columnCount() - 1  
 self.students.sortItems(last\_column\_index)  
  
 def retranslateUi(self, MainWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 MainWindow.setWindowTitle(\_translate("MainWindow", "Teacher"))  
 self.label\_3.setText(\_translate("MainWindow", "Великая цель образования-"))  
 self.label\_5.setText(\_translate("MainWindow", "Герберт Спенсер"))  
 self.add\_student.setText(\_translate("MainWindow", "Добавить студента"))  
 self.delete\_student.setText(\_translate("MainWindow", "Удалить студента"))  
 self.sorted\_alphabet.setText(\_translate("MainWindow", "Сортировать по алфавиту"))  
 self.sorted\_down.setText(\_translate("MainWindow", "Убыванию"))  
 self.label\_4.setText(\_translate("MainWindow", "Сортировать по:"))  
 self.sorted\_up.setText(\_translate("MainWindow", "Возрастанию"))  
 self.exit.setText(\_translate("MainWindow", "Выход"))  
 self.back.setText(\_translate("MainWindow", "Назад"))  
 self.label\_8.setText(\_translate("MainWindow", "это не знания, а действия"))  
 self.menu\_file.setTitle(\_translate("MainWindow", "Файл"))  
 self.menu\_information.setTitle(\_translate("MainWindow", "Информация"))  
 self.action\_save.setText(\_translate("MainWindow", "Сохранить"))  
 self.action\_open.setText(\_translate("MainWindow", "Открыть"))  
 self.action\_csv.setText(\_translate("MainWindow", "О csv-файлах"))

Окно студента

from PyQt6 import QtCore, QtGui, QtWidgets  
import sqlite3  
from PyQt6.QtCore import Qt  
  
  
class StudentLogic(object):  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(1005, 697)  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 Dialog.setFont(font)  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(0, 0, 1011, 701))  
 self.label.setText("")  
 self.label.setPixmap(QtGui.QPixmap("../../Downloads/1678348032\_bogatyr-club-p-tuchi-estetika-foni-pinterest-57.jpg"))  
 self.label.setObjectName("label")  
 self.line = QtWidgets.QFrame(parent=Dialog)  
 self.line.setGeometry(QtCore.QRect(620, 50, 20, 511))  
 self.line.setFrameShape(QtWidgets.QFrame.Shape.VLine)  
 self.line.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line.setObjectName("line")  
 self.line\_2 = QtWidgets.QFrame(parent=Dialog)  
 self.line\_2.setGeometry(QtCore.QRect(20, 30, 941, 16))  
 self.line\_2.setFrameShape(QtWidgets.QFrame.Shape.HLine)  
 self.line\_2.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line\_2.setObjectName("line\_2")  
 self.line\_3 = QtWidgets.QFrame(parent=Dialog)  
 self.line\_3.setGeometry(QtCore.QRect(20, 580, 941, 16))  
 self.line\_3.setFrameShape(QtWidgets.QFrame.Shape.HLine)  
 self.line\_3.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line\_3.setObjectName("line\_3")  
 self.sorted\_alphabet = QtWidgets.QPushButton(parent=Dialog)  
 self.sorted\_alphabet.setGeometry(QtCore.QRect(680, 220, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.sorted\_alphabet.setFont(font)  
 self.sorted\_alphabet.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.sorted\_alphabet.setObjectName("pushButton\_5")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(650, 60, 311, 61))  
 font = QtGui.QFont()  
 font.setPointSize(18)  
 self.label\_2.setFont(font)  
 self.label\_2.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_2.setObjectName("label\_2")  
 self.label\_3 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_3.setGeometry(QtCore.QRect(650, 110, 311, 31))  
 font = QtGui.QFont()  
 font.setPointSize(18)  
 self.label\_3.setFont(font)  
 self.label\_3.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_3.setObjectName("label\_3")  
 self.label\_4 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_4.setGeometry(QtCore.QRect(650, 150, 241, 31))  
 font = QtGui.QFont()  
 font.setPointSize(15)  
 self.label\_4.setFont(font)  
 self.label\_4.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_4.setText("")  
 self.label\_4.setObjectName("label\_4")  
 self.label\_5 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_5.setGeometry(QtCore.QRect(770, 150, 221, 31))  
 font = QtGui.QFont()  
 font.setPointSize(17)  
 self.label\_5.setFont(font)  
 self.label\_5.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_5.setObjectName("label\_5")  
 self.exit = QtWidgets.QPushButton(parent=Dialog)  
 self.exit.setGeometry(QtCore.QRect(120, 630, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.exit.setFont(font)  
 self.exit.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.exit.setObjectName("pushButton\_9")  
 self.pushButton\_10 = QtWidgets.QPushButton(parent=Dialog)  
 self.pushButton\_10.setGeometry(QtCore.QRect(370, 630, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.pushButton\_10.setFont(font)  
 self.pushButton\_10.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.pushButton\_10.setObjectName("pushButton\_10")  
 self.line\_4 = QtWidgets.QFrame(parent=Dialog)  
 self.line\_4.setGeometry(QtCore.QRect(660, 190, 301, 20))  
 self.line\_4.setFrameShape(QtWidgets.QFrame.Shape.HLine)  
 self.line\_4.setFrameShadow(QtWidgets.QFrame.Shadow.Sunken)  
 self.line\_4.setObjectName("line\_4")  
 self.label\_6 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_6.setGeometry(QtCore.QRect(640, 270, 171, 51))  
 font = QtGui.QFont()  
 font.setPointSize(13)  
 self.label\_6.setFont(font)  
 self.label\_6.setStyleSheet("color: rgb(255, 255, 255);")  
 self.label\_6.setObjectName("label\_6")  
 self.sorted\_down = QtWidgets.QPushButton(parent=Dialog)  
 self.sorted\_down.setGeometry(QtCore.QRect(680, 330, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.sorted\_down.setFont(font)  
 self.sorted\_down.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.sorted\_down.setObjectName("pushButton\_6")  
 self.label\_7 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_7.setGeometry(QtCore.QRect(890, 340, 61, 31))  
 self.label\_7.setText("")  
 self.label\_7.setPixmap(QtGui.QPixmap("../../Downloads/Стрелка вниз.png"))  
 self.label\_7.setObjectName("label\_7")  
 self.sorted\_up = QtWidgets.QPushButton(parent=Dialog)  
 self.sorted\_up.setGeometry(QtCore.QRect(680, 390, 281, 41))  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.sorted\_up.setFont(font)  
 self.sorted\_up.setStyleSheet("\n"  
" border-radius: 15px; \n"  
" border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.sorted\_up.setObjectName("pushButton\_11")  
 self.label\_8 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_8.setGeometry(QtCore.QRect(890, 400, 61, 31))  
 self.label\_8.setText("")  
 self.label\_8.setPixmap(QtGui.QPixmap("../../Downloads/Стрелка\_вверх.png"))  
 self.label\_8.setObjectName("label\_8")  
  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
  
 self.DataBase(Dialog)  
 self.calculateAverage()  
 self.exit.clicked.connect(self.close\_all\_windows)  
 self.sorted\_alphabet.clicked.connect(self.sort\_table\_by\_alphabet)  
 self.sorted\_up.clicked.connect(self.sort\_table\_by\_rating\_up)  
 self.sorted\_down.clicked.connect(self.sort\_table\_by\_rating\_down)  
 self.pushButton\_10.clicked.connect(Dialog.close)  
  
 def DataBase(self,Dialog):  
 self.students = QtWidgets.QTableWidget(Dialog)  
 self.students.setGeometry(QtCore.QRect(60, 50, 540, 520))  
 self.students.setObjectName("students")  
 self.students.setColumnCount(8)  
 self.students.setRowCount(0)  
 self.students.raise\_()  
 self.students.setHorizontalHeaderLabels(['ФИО'] + [str(i) for i in range(1, 6)] +["Экзаменациооная\nоценка"]+ ["Средний балл"])  
  
 *# Создаем пустой список для хранения фио* names = []  
  
 conn = sqlite3.connect("database\_students.db")  
 cur = conn.cursor()  
 *# Создаём таблицу, если еще не существует* cur.execute("""  
 CREATE TABLE IF NOT EXISTS students (  
 id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
 name TEXT,  
 mark1 INTEGER,  
 mark2 INTEGER,  
 mark3 INTEGER,  
 mark4 INTEGER,  
 mark5 INTEGER,  
 mark6 INTEGER);  
 """)  
  
 for name in names:  
 *# Здесь мы сравниваем значение столбца name таблицы "students" с переданным значением name переменной* cur.execute("""INSERT OR IGNORE INTO students (name)  
 SELECT ?  
 WHERE NOT EXISTS (  
 SELECT name FROM students WHERE name = ?);""", (name, name,))  
 *# Выбираем все значения из столбца name и сохранем в переменную names* cur.execute("SELECT name FROM students")  
  
 cur.execute("SELECT name, mark1, mark2, mark3, mark4, mark5, mark6 FROM students")  
 rows = cur.fetchall()  
  
 self.students.setRowCount(len(rows))  
 for row, data in enumerate(rows):  
 for col, value in enumerate(data):  
 if value is not None:  
 item = QtWidgets.QTableWidgetItem(str(value))  
 else:  
 item = QtWidgets.QTableWidgetItem("") *# Создаем пустой элемент ячейки* self.students.setItem(row, col, item)  
 self.students.resizeColumnsToContents()  
 conn.commit()  
 conn.close()  
  
 def calculateAverage(self):  
 *# Вычисление среднего значения для каждой строки таблицы* for row in range(self.students.rowCount()):  
 total\_sum = 0  
 total\_count = 0 *# Общее количество значений* for col in range(1, 8): *# Проход по столбцам с оценками* item = self.students.item(row, col) *# Получение значения ячейки* if item is not None and item.text():  
 value = float(item.text())  
 total\_sum += value  
 total\_count += 1  
 if total\_count != 0: *# Проверка наличия значений* average = total\_sum / total\_count  
 last\_column\_index = self.students.columnCount() - 1  
 average\_item = QtWidgets.QTableWidgetItem(f"{average:.2f}") *# Создание ячейки для среднего значения* self.students.setItem(row, last\_column\_index,  
 average\_item) *# Установка среднего значения в ячейку последнего столбца* def close\_all\_windows(self):  
 app = QtWidgets.QApplication.instance()  
 for widget in app.topLevelWidgets():  
 if widget.isVisible():  
 widget.close()  
  
 def sort\_table\_by\_alphabet(self, column):  
 self.students.sortItems(column)  
  
 def sort\_table\_by\_rating\_down(self, column):  
 last\_column\_index = self.students.columnCount() - 1  
 self.students.sortItems(last\_column\_index, Qt.SortOrder.DescendingOrder)  
  
 def sort\_table\_by\_rating\_up(self, column):  
 last\_column\_index = self.students.columnCount() - 1  
 self.students.sortItems(last\_column\_index)  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "Student"))  
 self.sorted\_alphabet.setText(\_translate("Dialog", "Сортировать по алфавиту"))  
 self.label\_2.setText(\_translate("Dialog", "Великая цель образования-"))  
 self.label\_3.setText(\_translate("Dialog", "это не знания, а действия"))  
 self.label\_5.setText(\_translate("Dialog", "Герберт Спенсер"))  
 self.exit.setText(\_translate("Dialog", "Выход"))  
 self.pushButton\_10.setText(\_translate("Dialog", "Назад"))  
 self.label\_6.setText(\_translate("Dialog", "Сортировать по:"))  
 self.sorted\_down.setText(\_translate("Dialog", "Убыванию"))  
 self.sorted\_up.setText(\_translate("Dialog", "Возрастанию"))

О csv-файлах

from PyQt6 import QtCore, QtGui, QtWidgets  
  
  
class AboutCSV(object):  
 def setupUi(self, Dialog):  
 Dialog.setWindowIcon(QtGui.QIcon('graduation-hat.png'))  
 Dialog.setObjectName("Dialog")  
 Dialog.resize(553, 619)  
 Dialog.setStyleSheet("background-color: rgb(248, 255, 230);")  
 self.label = QtWidgets.QLabel(parent=Dialog)  
 self.label.setGeometry(QtCore.QRect(30, 210, 491, 371))  
 self.label.setText("")  
 self.label.setPixmap(QtGui.QPixmap("../../Desktop/Курсовой проект/About\_CSV.png"))  
 self.label.setObjectName("label")  
 self.label\_2 = QtWidgets.QLabel(parent=Dialog)  
 self.label\_2.setGeometry(QtCore.QRect(30, 0, 491, 231))  
 self.label\_2.setText("")  
 self.label\_2.setPixmap(QtGui.QPixmap("../../Desktop/Курсовой проект/Example\_CSV.png"))  
 self.label\_2.setObjectName("label\_2")  
 self.back = QtWidgets.QPushButton(parent=Dialog)  
 self.back.setGeometry(QtCore.QRect(220, 570, 151, 41))  
 font = QtGui.QFont()  
 font.setPointSize(12)  
 self.back.setFont(font)  
 self.back.setStyleSheet(" border-radius: 15px; \n"  
"border: 2px solid #094065;\n"  
"background-color: qlineargradient(spread:pad, x1:0.9899, y1:0.402, x2:0.930348, y2:0.932, stop:0 rgba(202, 214, 229, 255), stop:0.830846 rgba(254, 237, 206, 255));")  
 self.back.setObjectName("pushButton")  
  
 self.retranslateUi(Dialog)  
 QtCore.QMetaObject.connectSlotsByName(Dialog)  
 *# Закрываем окно при нажатии на кнопку* self.back.clicked.connect(Dialog.close)  
  
 def retranslateUi(self, Dialog):  
 \_translate = QtCore.QCoreApplication.translate  
 Dialog.setWindowTitle(\_translate("Dialog", "About csv"))  
 self.back.setText(\_translate("Dialog", "Назад"))

**Список используемой литературы:**

1. "Mastering GUI Programming with Python: Develop impressive cross-platform GUI applications with PyQt" by Burkhard Meier - Амазон: https://www.amazon.com/Mastering-GUI-Programming-Python-cross-platform/dp/1789612903.

2. "PyQGIS Programmer's Guide: Extend QGIS using Python 3" by Gary Sherman - Амазон: https://www.amazon.com/PyQGIS-Programmers-Guide-Extend-Python-ebook/dp/B01EZA6ZR4.

3. Языки программирования. Python : учебно-методическое пособие для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» : в 2 ч. / В. В. Иванченко [и др.] ; Белорусский национальный технический университет, Кафедра "Программное обеспечение информационных систем и технологий". – Минск : БНТУ, 2021. – Ч. 1. – 91 с.

4. GUI Programming with Python: Develop Interactive and Responsive Desktop Applications" by Alan D. Moore